LAPORAN PRAKTIKUM DATA STRUCTURE

ARRAYS

Dosen Pengampu:

H. Fatchurrochman,M.Kom

Asisten Praktikum:

Fillah Anjany 230605110033

Oleh :

Muhammad Alif Mujaddid **240605110082**

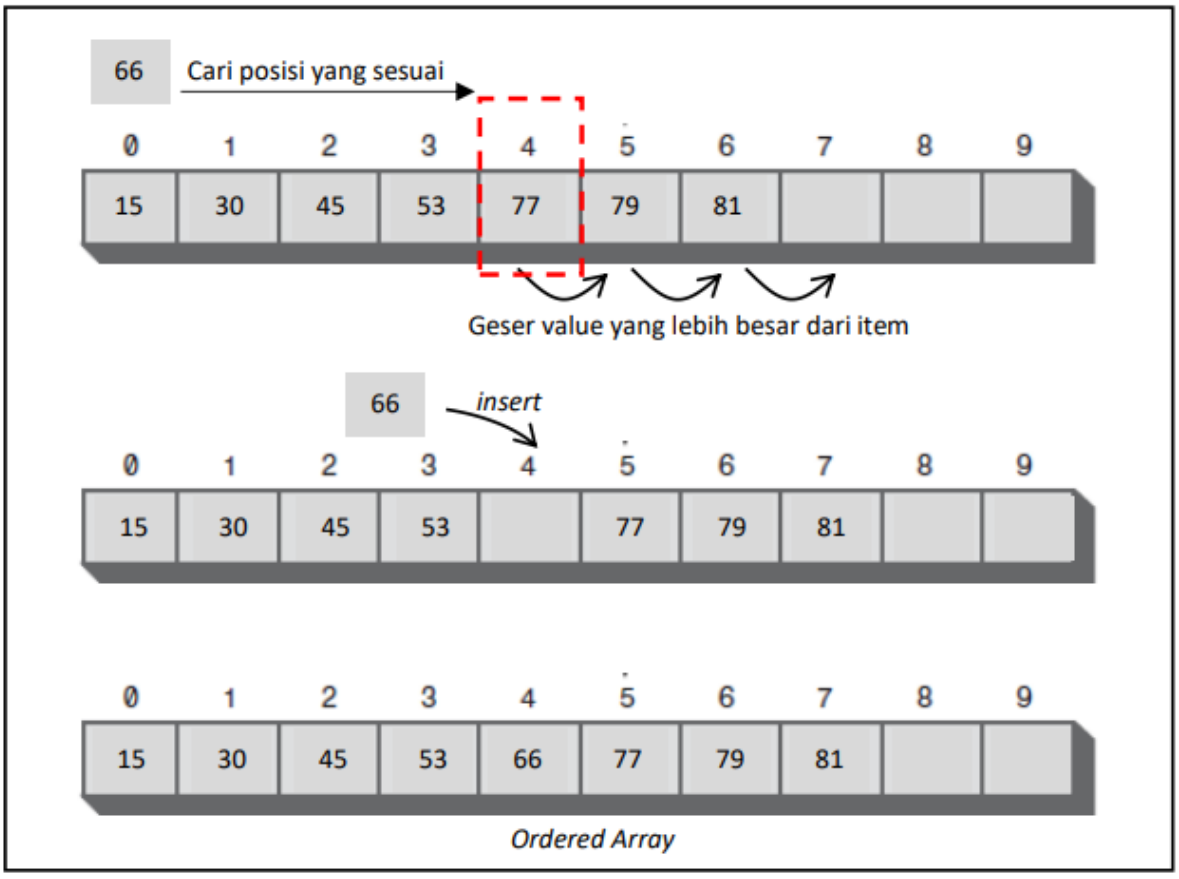
Jurusan Teknik Informatika Fakultas Sains dan Teknologi

UIN Maulana Malik Ibrahim Malang

2025

A. PRAKTIKUM

1. Pada listing nomer 5 (tugas pendahuluan), method insert digunakan untuk menambahkan item pada cell yang belum terisi tanpa memberhatikan value item yang ditambahkan sehingga elemen pada array disimpan secara tidak berurutan (unordered). Agar item dapat disimpan pada urutan yang sesuai dengan value-nya maka perlu dilakukan pencarian posisi cell yang tepat bagi item yang akan dimasukkan dengan cara membandingkan tiap item pada cell dengan item yang akan dimasukkan, yaitu pencarian secara linier. Setelah cell tepat ditemukan, langkah selanjutnya adalah menyiapkan cell tersebut untuk diisi jika sudah ada item yang tersimpan pada cell tersebut. Hal ini bisa dilakukan dengan cara menggeser item yang memiliki value lebih besar dari item yang akan dimasukkan, dengan demikian terdapat cell kosong untuk diisi dengan item baru. Langkah-langkah insert item pada ordered array ditunjukkan pada Gambar 1.1 berikut ini.



Gambar 1.1 Langkah *insert* item pada *ordered array*

Tuliskan listing untuk method insert untuk menyimpan elemen array secara berurutan (ordered)!

```java
public void insert(int value){
    int i ;
    for (i = 0; i < nElemen; i++) {
        if(arr[i] > value){
            break;
        }
    }
    for (int j = nElemen; j >i; j--) {
        arr[j] = arr[j-1];
    }
    arr[i] = value;
    // arr[nElemen] = value;
    nElemen++;
}
```

```java
arr.insert(value:1);
arr.insert(value:100);
```

Output:

```
1 30 40 55 70 75 80 85 90 100
```

Kode lengkap : https://github.com/addid-cloud/dsa/blob/main/Praktikum/HighArrayApp.java

B. Kesimpulan

1. Unordered Arrays dan Ordered Arrays

Unordered Arrays (Array Tidak Terurut) Seperti namanya, elemen dalam array ini tidak diatur dalam urutan tertentu. Ketika Anda ingin menambahkan data baru, prosesnya sangat cepat dan efisien. Anda cukup meletakkannya di akhir array pada indeks kosong pertama yang tersedia. Operasi ini memiliki kompleksitas waktu konstan, atau O(1), karena tidak peduli seberapa besar array tersebut, penyisipan di akhir selalu memakan waktu yang sama.

Namun, kelemahan utamanya muncul saat pencarian data. Karena tidak ada pola atau urutan, satu-satunya cara untuk menemukan elemen tertentu adalah dengan melakukan *pencarian linear*, yaitu memeriksa setiap elemen satu per satu dari awal hingga akhir. Proses ini bisa sangat lambat jika array berisi banyak data (kompleksitas waktu O(N)).

Dalam array terurut / ordered list, setiap elemen disimpan berdasarkan urutan nilainya, baik itu secara menaik (*ascending*) maupun menurun (*descending*). Keunggulan utamanya adalah efisiensi pencarian data. Karena datanya terurut, kita bisa menggunakan algoritma yang jauh lebih cepat seperti *binary search* (pencarian biner). Akan tetapi, keunggulan ini harus dibayar dengan proses penyisipan data yang lebih lambat. Untuk menambahkan elemen baru, program harus terlebih dahulu menemukan posisi yang tepat agar urutan array tetap terjaga. Setelah posisi ditemukan, semua elemen yang nilainya lebih besar harus digeser untuk memberikan ruang bagi elemen baru tersebut. Proses penggeseran ini membuat penyisipan menjadi kurang efisien (kompleksitas waktu O(N)) dibandingkan dengan array tidak terurut. Hal yang sama berlaku untuk penghapusan data, di mana "lubang" yang ditinggalkan oleh elemen yang dihapus harus ditutup dengan menggeser elemen lainnya.

2. Linear Search dan Binary Search

Linear Search Ini adalah metode pencarian yang paling dasar dan intuitif. Algoritma ini bekerja dengan cara memeriksa setiap elemen dalam array secara berurutan, mulai dari indeks pertama (indeks 0) hingga elemen tersebut ditemukan atau hingga seluruh array selesai diperiksa.

Binary Search Metode ini jauh lebih canggih dan efisien, namun memiliki syarat mutlak: hanya bisa digunakan pada array yang sudah terurut (ordered array). Algoritma ini dimulai dengan memeriksa elemen tengah.

1. Jika elemen tengah adalah data yang dicari, pencarian selesai.
2. Jika data yang dicari lebih kecil dari elemen tengah, maka pencarian dilanjutkan hanya pada paruh kiri array.
3. Jika data yang dicari lebih besar dari elemen tengah, maka pencarian dilanjutkan hanya pada paruh kanan array. Proses ini diulang terus-menerus, membagi rentang pencarian menjadi setengahnya di setiap langkah.

3. Menyimpan Objek (Storing Object)

Intinya, array itu ada dua tipe: yang berantakan (unordered) dan yang rapi (ordered). Kalo yang berantakan, nambahin data baru itu cepet banget, tinggal taruh di paling akhir. Tapi pas mau nyari data, jadi lama karena harus dicek satu-satu dari awal (ini namanya *linear search*). Nah, kalo array yang rapi, nyari datanya super cepet pake trik potong kompas (namanya *binary search*), tapi jadi ribet pas mau nambahin data baru karena harus geser-geser data lain biar tetap urut. Terus, kerennya lagi, array itu nggak cuma buat nyimpen angka atau huruf doang, tapi bisa juga buat nyimpen "objek". Anggap aja objek itu kayak profil lengkap seorang Mahasiswa, yang isinya ada NIM, nama, dan alamat. Jadi, satu array bisa nampung data profil banyak siswa sekaligus.